Amendments to the Claims:

This listing of claims will replace all prior versions, and listings of claims in the application:

Listing of Claims:

1. (Original) A method of managing a memory located on a peripheral device of a computing system, the method comprising:

in response to a first request, dynamically allocating a first portion of the memory, thereby making the first portion of the memory accessible to a co-processor of the peripheral device;

in response to a second request, dynamically mapping the first portion of the memory to one or more virtual addresses in a first region of a virtual memory space, thereby making the first portion of the memory directly accessible to a process executing on a central processing unit (CPU) of the computing system,

wherein the first portion of the memory is dynamically unmappable in response to an unmapping request.

2. (Original) The method of claim 1, wherein the first and second requests are generated by a driver program for the peripheral device.

3. (Original) The method of claim 1, wherein the second request occurs at a time other than during system initialization.

4. (Original) The method of claim 1, wherein the peripheral device includes a graphics card.

5. (Original) The method of claim 4, wherein the first portion of the memory corresponds to a frame buffer for storing pixel data of an image.

6. (Original) The method of claim 4, wherein the first portion of the memory corresponds to a command buffer for queuing commands to be executed by a graphics co-processor located on the graphics card.

7. (Original) The method of claim 1, further comprising:

in response to a third request, dynamically mapping a second portion of the memory to one or more virtual addresses in a second region of the virtual memory space.

8. (Original) The method of claim 7, wherein the first and second portions of the memory have at least one memory location in common.

9. (Original) The method of claim 8, wherein the first and second regions of the virtual memory space do not have any virtual addresses in common.

10. (Original) The method of claim 7, wherein the first and second portions of the memory are non-contiguous and the first and second regions of the virtual memory space are contiguous.

11. (Original) The method of claim 1, further comprising:

in response to the second request, modifying one or more page table entries to establish an association between the one or more virtual address and a location in the first portion of the memory.

12. (Original) The method of claim 1, wherein the virtual memory space includes a kernel space and a user space.

13. (Original) The method of claim 12, wherein the first region of the virtual memory space is in the kernel space.

14. (Original) The method of claim 12, wherein the first region of the virtual memory space is in the user space.

15. (Original) The method of claim 12, further comprising:

prior to the act of dynamically mapping the first portion of the memory, determining from the second request whether the first portion of the memory is to be mapped into the kernel space or the user space.

16. (Original) The method of claim 1, further comprising:

in response to a third request that includes an allocation request and a mapping request:

allocating a second portion of the memory from a heap; and

after allocating the second portion of the memory, dynamically mapping the memory to one or more virtual addresses in a second region of the virtual address space.

17. (Original) The method of claim 1, further comprising:

in response to a third request, dynamically unmapping the first portion of the memory.

18. (Original) The method of claim 17, further comprising:

in response to the second request, modifying one or more page table entries to establish an association between the one or more virtual address and a location in the first portion of the physical memory; and

in response to the third request, modifying the one or more page table entries such that the one or more the virtual addresses are no longer associated with the locations.

19. (Original) The method of claim 17, wherein the third request is generated by a driver program for the peripheral device.

20. (Original) The method of claim 17, wherein the third request includes a deallocation request, the method further comprising:

in response to the third request, deallocating the first portion of the memory, thereby returning the first portion of the memory to a heap,

wherein deallocation occurs after the act of unmapping the first portion of the memory.

21. (Original) The method of claim 1, further comprising:

in response to a third request, the third request being a request to map a second portion of the memory:

determining whether the second portion of the memory matches the first portion of the memory;

in the event that the second portion of the memory matches the first portion of the memory, reusing the mapping of the first portion of the memory to satisfy the third request; and

in the event that the second portion of the memory does not match the first portion of the memory, mapping the second portion of the memory to one or more virtual addresses in a second region of the virtual memory space.

22. (Original) The method of claim 21, wherein the second portion of the memory matches the first portion of the memory in the event that every location within the second portion of the memory is also within the first portion of the memory.

23. (Currently amended) A computer program product comprising:

a tangible computer readable medium encoded with program code, the program code including:

program code for generating an allocation request;

program code for responding to the allocation request by dynamically allocating a first portion of a memory of a peripheral device, thereby making the first portion of the memory accessible to a co-processor of the peripheral device;

program code for generating a mapping request;

program code for responding to the mapping request by dynamically mapping the first portion of the memory to one or more virtual addresses in a first region of a virtual memory space, thereby making the first portion of the memory directly accessible to a process executing on a central processing unit (CPU) of a computing system;

program code for generating an unmapping request; and

program code for responding to the unmapping request by unmapping the portion of the memory, thereby making the portion of memory no longer accessible to the process.

24. (Original) The computer program product of claim 23, wherein the computer readable medium comprises a magnetic storage medium encoded with the program code.

25. (Original) The computer program product of claim 23, wherein the computer readable medium comprises an optical storage medium encoded with the program code.

26. (Original) The computer program product of claim 23, wherein the computer readable medium comprises a carrier signal encoded with the program code and adapted for transmission via a network.

27. (Original) A product for use with a computing system, the product comprising:

a peripheral device adapted to be installed in the computing system; and

a computer readable medium encoded with driver program code for a driver of the peripheral device, the driver program code including:

program code for generating an allocation request;

program code for responding to the allocation request by dynamically allocating a first portion of a memory of a peripheral device, thereby making the first portion of the memory accessible to a co-processor of the peripheral device;

program code for generating a mapping request;

program code for responding to the mapping request by dynamically mapping the first portion of the memory to one or more virtual addresses in a first region of a virtual memory space, thereby making the first portion of the memory directly accessible to a process executing on a central processing unit (CPU) of the computing system;

program code for generating an unmapping request; and

program code for responding to the unmapping request by unmapping the portion of the memory, thereby making the portion of memory no longer accessible to the process.